AD-P000 112

UNFORCED IMAGE PARTITIONING BY WEIGHTED PYRAMID LINKING

Tsai-Hong Hong
Azriel Rosenfeld

Computer Vision Laboratory, Computer Science Center
University of Maryland, College Park, MD 20742

ABSTRACT

This paper describes a method of image segmentation that creates a partition of the image into compact, homogeneous regions using a parallel, iterative approach that does not require immediate forced choices. The approach makes use of a "pyramid" of successively reduced-resolution versions of the image. It defines link strengths between pairs of pixels at successive levels of this pyramid, based on proximity and similarity, and iteratively recomputes the pixel values and adjusts the link strengths. After a few iterations, the link strengths stabilize, and the links that remain strong define a set of subtrees of the pyramid. Each such tree represents a compact (piece of a) homogeneous region in the image; the leaves of the subtree are the pixels in the region, and the size of the region depends on how high the root of the tree lies in the pyramid. Thus the trees define a partition of the image into (pieces of) homogeneous regions.

1. Introduction

Most of the existing methods of image segmentation [1,2] are based on forced-choice decisions. In methods that classify pixels into subpopulations, we must decide to which class each pixel belongs. In methods that partition the image into homogeneous regions using splitting and merging processes, we must decide, for each current region, whether to split it, or whether to merge it with a neighboring region (and if so, with which one). This forced-choice aspect of segmentation is undesirable, since many of the decisions may be wrong, particularly when they are made on the basis of very little information, and it is difficult to undo the effects of wrong decisions.

In segmentation by pixel classification, a "relaxation" approach [3] can be used to defer the classification decisions until more information is available. In this approach we compute a degree of membership for each pixel in each class, or a "probability" that it belongs to each class; and we then iteratively adjust these membership values, based on the values at neighboring pixels and the compatibilities of the various possible combinations of class memberships of pairs of neighbors. After a few iterations, the membership values stabilize, with some values becoming or remaining relatively high and others becoming very low, so that it becomes easy to make the final classification decisions.

Segmentation by partitioning into homogeneous regions - e.g., regions of approximately constant value - is generally more powerful than segmentation by pixel classification, because the information on which it is based is computed over regions rather than ("myopically") over small neighborhoods of pixels. Thus it would be desirable to develop a region-based segmentation scheme in which decisions are not made immediately. This paper defines such a scheme and gives examples of the results obtained when it is applied to various types of images. Section 2 describes the general principles of this scheme and compares it with some related approaches; Section 3 discusses the algorithm; and Section 4 presents experimental results.

2. Weighted pyramid linking

Our approach to unforced image partitioning makes use of a "pyramid" of successively reduced-resolution versions of the given image, say of sizes 2^n by 2^n, 2^{n-1} by 2^{n-1},..., 2x2. The base of the pyramid (level 0) is the input image, and each successive level is constructed by averaging 4 by 4 blocks of pixels on the level below, where the blocks overlap 50% in x and in y. (For convenience, each level is regarded as cyclically closed, so that its top row is adjacent to its bottom row and its left column to its right column.) Thus each pixel on a given level has 16 "sons" on the level below (if any) that contribute to its average, and 4 "fathers" on the level above (if any) to whose average it contributes. This type of pyramid has also been used for segmentation purposes by other investigators; e.g., see the work of Hanson and Riseman described in [4].

The basic idea in our approach is to define link strengths between "neighboring" pixels (i.e., father/son pairs) on adjacent levels of the pyramid, based on the similarity (in value) and proximity (in (x,y) coordinates) of each such pair. We then recompute the pixel values (at the levels above the base) as weighted averages of their sons'

The support of the Defense Advanced Research Projects Agency and the U.S. Army Night Vision Laboratory under Contract DAAG-53-76C-0138 (DARPA Order 3206) is gratefully acknowledged, as is the help of Clara Robertson in preparing this paper.

values, where the weights depend on the link strengths. These new values define new link strengths, and the process is iterated. (The details of the algorithm will be given in the next section.) After a few iterations, the link strengths stabilize, and the links that remain strong define subtrees of the pyramid. As it turns out, each such tree defines a compact homogeneous region in the image, where the leaves of the tree are the pixels belonging to the region, and the height of the tree corresponds to the region size (the larger the region, the higher the root of its tree lies in the pyramid). Thus the weighted links can be used to define a partition of the image into compact homogeneous regions. Note that this partition is not defined immediately, but only after the link weights have stabilized.

To see intuitively why this approach should work, consider the case of a homogeneous compact region on a homogeneous background. Pixels in the interior of the region (or background) will link strongly to all their fathers, since these fathers' values are averages of image blocks that lie in the same region! A pixel near the region border, however, will link more strongly to a father that lies inside the region than to one that lies partly outside, since it is more similar in value to the former. Thus when we recompute the fathers' values, a father whose image block lies mostly inside the region will get closer in value to the average of the region, since it is more strongly linked to its sons that lie in the region than to those that lie in the background; and conversely. This makes its links to the former sons even stronger, and to the latter even weaker, so that the link strengths and values should converge. Now consider a pixel whose block lies mostly inside the region, but whose fathers' blocks all lie mostly outside, because they are bigger than the region. By the argument just given, the pixel's value should tend toward the region average, while its fathers' values should tend toward the background average, so that the pixel does not remain strongly linked to any of its fathers, and becomes the root of a tree representing (a compact portion of) the region.

It is of interest to compare this approach to some earlier segmentation schemes based on pyramid linking or on link strengths. In [5-6] link strengths are computed between each father/son pair, but we keep only the strongest of the four links between a pixel and its fathers. We then recompute the pixel values allowing only those sons that are linked to a pixel to contribute to its value; recompute the link strengths based on these new values; and iterate the process. Note that in this scheme every pixel must link to one of its fathers; thus the links define precisely four trees, rooted at the top (2x2) level, so that the image is segmented into precisely four sets of pixels. These sets do not correspond to compact regions, but do tend to correspond to homogeneous subpopulations of pixels. Thus the segmentation scheme of [1-2] is more like a pixel clustering and classification scheme than an image partitioning scheme; and it also makes forced choices immediately, since it keeps only the strongest upward link from each pixel. Extensions of this scheme to segmentation based on color or texture, and to waveform or contour segmemtation, are described in [7-10].

A pyramid linking method which does make use of all the link strengths, rather than discarding all but the strongest upward link, is described in [11]. However, in this method the link strengths are normalized so that they sum to 1; thus here too the links are forced to extend upward from every pixel (divided among its fathers appropriately) all the way to the top level. In fact, the link strengths tend to converge to 0 or 1 where the process is iterated, so that this method too defines a segmentation of the image into four subpopulations of pixels, rather than a partition into regions.

A weighted pixel linking scheme not involving a pyramid is described in [12]. Here a link strength is computed for each pair of neighboring pixels based on their closeness in value. The image is then smoothed by replacing each pixel with the average of its neighbors, weighted by their link strengths. Using these new values, the link strengths are recomputed, and the process is iterated. This tends to produce a very high-quality smoothing, and the links that remain strong could be used to define a segmentation of the image into homogeneous regions; but this method would not always be reliable, since it is based on small neighborhoods. The method defined in this paper is analogous to the scheme in [12], but using "vertical" links (between fathers and sons) in a pyramid, rather than "horizontal" links (between brothers) in an image at a single resolution. Our method could be generalized to make use of horizontal as well as vertical link strengths, but we shall not pursue this possibility here.

3. The algorithm

The algorithm is initialized, as mentioned earlier, by building the pyramid using unweighted averaging of 4x4 blocks that overlap 50% horizontally and vertically. Alternatively, we could use nonoverlapping 2x2 blocks (for the initialization only; a pixel still has 16 sons in the subsequent steps), or we could use the median instead of the mean; but these variations were found to make little difference in the results.

Let $v(P)$ denote the value of pixel P in the pyramid, say on level ℓ. Initially, if $\ell=0$ this is the gray level of an input pixel, and if $\ell>0$ it is the mean of the values of P's 16 sons. Let $\sigma(P)$ be the standard deviation of these sons' values (or if $\ell=0$, we take σ to be a constant; we used 5 in our experiments).

Let P^* be one of the fathers of P. The link strength between P and P^* is defined by

$$w(P,P^*) \equiv (1+d(P,P^*))\frac{\exp(-\frac{1}{2}[\frac{v(P)-v(P^*)}{\sigma(P)}]^2)}{\sqrt{2\pi}\;\sigma(P)}$$

In this expression, the first factor depends on the distance between (the centers of) P and P*; d is taken to be 3 for the closest father, 1 for the farthest, and $\sqrt{5}$ for the other two. (It can be verified that these are proportional to the Euclidean distances between the centers.) This factor makes the sets of pixels that belong to a given tree more compact; if it is omitted, these sets become more irregular in shape. The factors $\frac{1}{\sigma(P)}$ reflect the (non) variability of the sons of P; if they are highly variable, P does not link strongly to any of its fathers. Finally, the exp factor depends on the similarly in value of P and P*; if they are very dissimilar, the link is weak.

We now want to recompute the pixel values at levels >0 as weighted averages of their sons' values, where the weights depend on the link strengths. Note first that the weight given to a son must also depend on the (weighted) "area" of the image represented by that son; for example, if one son had unit strength links (down through successive levels) to a single image pixel, and zero strengths to all its other descendants, we would not want to give it as much weight as a son that had high-strength links to many image pixels. Let a(P) be the "area" of pixel P; initially, for a pixel at level ℓ, we have $a(P) = 2^{2\ell}$, since P represents a 2^ℓ by 2^ℓ image block. Subsequently, let a(P') be the area of a son P' of P, and let w(P',P) be the link strength between them. Then $a(P) = \sum_{P'} w(P',P)a(P')/W(P')$ (where the sum is over the sons of P' of P); here $W = \sum_{P'*} w(P',P*)$ (the sum being over the fathers P'* of P'). Note that in computing a(P) we are actually using normalized weights, i.e., $\sum w(P',P'*)/W=1$. This is because it seems reasonable that the "area" of a pixel should be distributed among its fathers in a normalized fashion, in order to insure that the total "area" of all pixels at a given level remains equal to the area of the image.

Finally, the new value of pixel P is given in terms of its sons' values by

$$v(P) = \frac{\sum_{P*} v(P')a(P')w(P',P)}{\sum_{P'} a(P')w(P',P)}$$

where the sums are over the sons P' of P. Similarly, the new standard deviation is given by

$$\sigma(P) = \sqrt{\frac{\sum_{P'} (v(P) - v(P'))^2 a(P')w(P',P)}{\sum_{P'} a(P')w(P',P)}}$$

The process is iterated; in our experiments, only two or three iterations were necessary.

After the desired number of iterations, we call a pixel a "root" if it is on the top level (2x2), or if the sum of its link strengths to all its fathers is negligible (in our experiments: $\leq 10^{-5}$). The nonroot pixels are then assigned to trees by using only their most strongly linked fathers.

4. Experiments

The algorithm just described was applied to the three images shown in Figure 1: photomicrographs of some chromosomes (right) and blood cells (left), and an infrared image of a tank. Each image is 64x64 pixels; thus the top (2x2) level of the pyramid is level 5. At each iteration, the gray level displayed for each pixel is the value at the root of its tree. We see that even after a single iteration, the trees define a decomposition of the image into regions having a small set of values; and in one or two more iterations the set of values is reduced even further.

Table 1 lists the root nodes at each level, and their values, for each image for as many iterations as were needed until there was no further change in the set of roots. We see that the more complex the image, the more iterations are required for the set of roots to stabilize; but that even for the most complex image, the changes after the first two or three iterations have little effect on the segmentation of the image.

Figure 2 shows printouts of the displayed images after the first (parts a-c) and last (parts d-f) iterations, where the value printed in each region identifies the root of the tree to which it belongs; the digit is the level, and the letters are used to distinguish the roots at that level. We see that after a few iterations, the leaves of each tree define a small set of compact regions. As Table 1 indicates, regions that are compact pieces of a single homogeneous region have nearly the same value. Note that because of the coordinate wraparound, regions on opposite sides of the image may belong to the same tree.

5. Concluding remarks

We have exhibited a method of segmenting an image into compact honogeneous regions by constructing links between "neighboring" pixels at consecutive levels of a "pyramid".

An important feature of this method is that each region is represented by a tree having the pixels of the region as leaves. The height of this tree is proportional to the log of the region size. Thus, even for large regions, all the pixels in the region are relatively closely linked to the root of the tree, and hence to each other. The pyramid structure makes it possible for information to propagate between different parts of a region relatively rapidly. Moreover, the root of the tree can be used as a node to represent the region in various region-level relational structures. Thus the tree constitutes a transition between the pixel-level representation of the region and more abstract representations.

Another important feature of our method is that the trees are produced by a cooperative process in which link strengths are iteratively adjusted. Under this process, root pixels representing regions become easy to recognize, because their link strengths to their fathers all become negligible. They are harder to recognize in the original pyramid, where the pixels (especially at higher levels) represent mixtures of image pixels, so that the link strengths are not initially negligible.

Image processing and segmentation techniques based on "local" operations performed in a pyramid can be implemented very rapidly in parallel on a tree-structured cellular processor [13]. It is possible that processes of this type also play a role in biological visual systems, where the input image is represented at a range of resolutions [14].

REFERENCES

1. A Rosenfeld and A. C. Kak, Digital Picture Processing, second edition, Academic Press, New York, 1982, Chapter 10.

2. T. Pavlidis, Structural Pattern Recognition, Springer, New York, 1977.

3. A. Rosenfeld and L. S. Davis, Cooperating processes for low-level vision: a survey, Artificial Intelligence 17, 1981, 245-263.

4. A. Klinger and S. L. Tanimoto, Structured Computer Vision, Academic Press, New York, 1980.

5. P. J. Burt, T. T. Hong and A. Rosenfeld, Image segmentation and region property computation by cooperative hierarchical computation, IEEE Trans. Systems, Man, Cybernetics 11, 1981, 802-809.

6. T. Silberberg, S. Peleg, and A. Rosenfeld, Multi-resolution pixel linking for image smoothing and segmentation, TR-977, Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD, November 1980.

7. T. H. Hong and A. Rosenfeld, Multiband pyramid linking, TR-1025, Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD, March 1981.

8. M. Pietikäinen and A. Rosenfeld, Image segmentation by texture using pyramid node linking, IEEE Trans. Systems, Man, Cybernetics, 11, 1981, 822-825.

9. M. Pietikäinen, A. Rosenfeld, and I. Walter, Split-and-link algorithms for image segmentation, Pattern Recognition 14, 1982, in press.

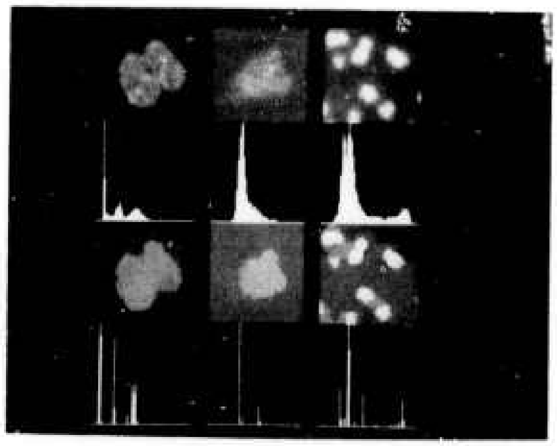
10. K. A. Narayanan and A. Rosenfeld, Approximation of waveforms and contours by one-dimensional pyramid linking, Pattern Recognition 14, 1982, in press.

11. K. A. Narayanan, S. Peleg, A. Rosenfeld, and T. Silberberg, Iterative image smoothing and segmentation by weighted pyramid linking, TR-989, Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD, December 1980.

12. J. O. Eklundh and A. Rosenfeld, Image smoothing based on neighbor linking, IEEE Trans. Pattern Analysis Machine Intelligence 3, 1981, 670-683.

13. C. R. Dyer. A VLSI pyramid machine for hierarchical parallel image processing, Proc. PRIP '81, August 1981, 381-386.

14. L. Uhr, Psychological motivation and underlying concepts, in [4], 1-30.



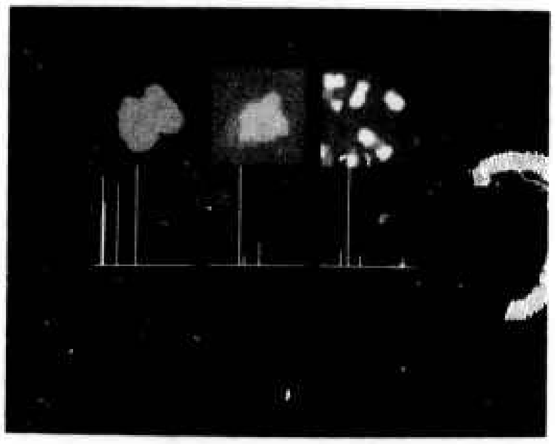


Figure 1. a) Input images and their histograms
b) Results after one iteration
c) Results after last iteration

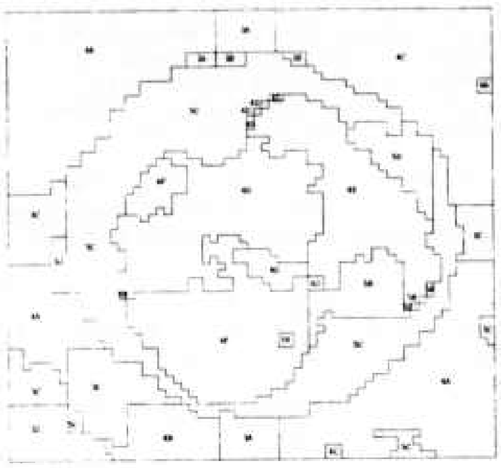


Figure 2a

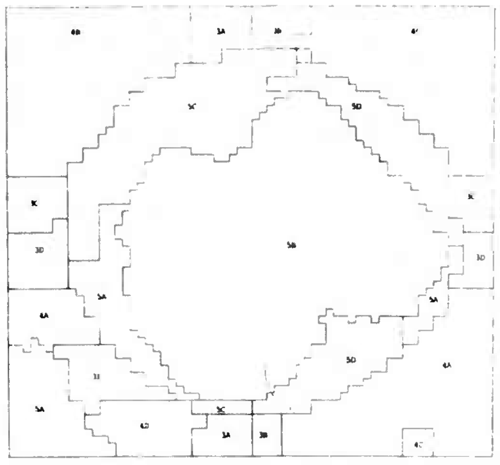


Figure 2d

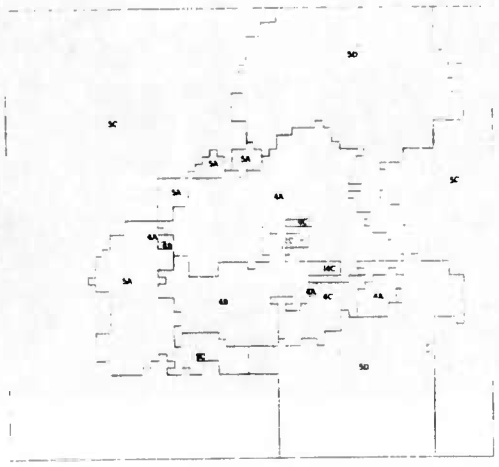


Figure 2b

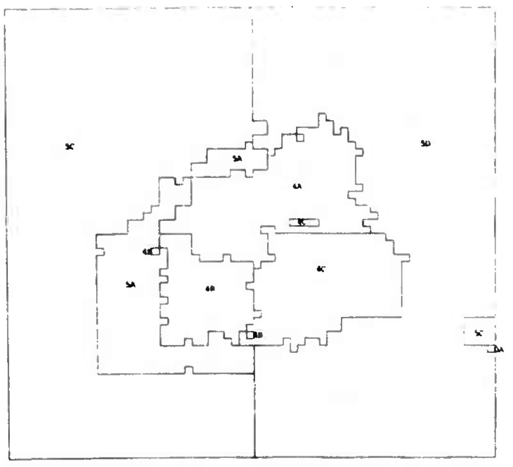


Figure 2e

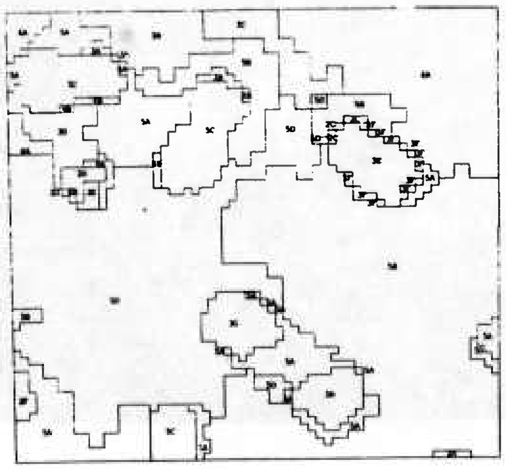


Figure 2c

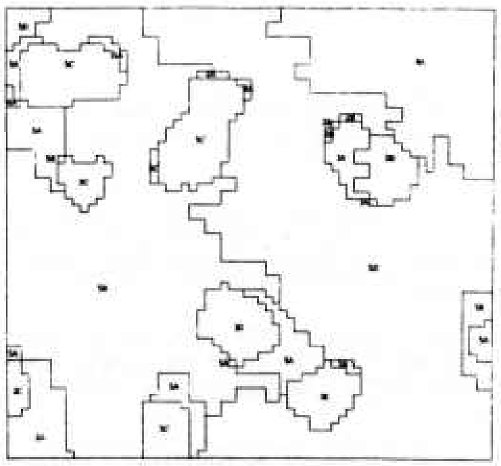


Figure 2f

Table 1. Root nodes and their values at each iteration for the three images. The root labels (see Figure 2) are given only for the first and last iterations.

(a) Cell image; there were no changes in the set of root nodes after the third iteration.

Iteration	Level	Root Label	Coordinates	Value
1	3	A	(0,3)	5.65
		B	(0,4)	6.87
		C	(3,0)	5.68
		D	(4,0)	6.08
		E	(6,1)	5.85
	4	A	(3,3)	5.56
		B	(0,0)	5.22
		C	(0,3)	5.34
		D	(1,1)	26.89
		E	(1,2)	28.27
		F	(2,1)	29.14
		G	(2,2)	29.64
	5	A	(1,0)	15.63
		B	(1,1)	23.96
		C	(0,0)	15.28
		D	(0,1)	15.61
2	3		(0,3)	5.34
			(0,4)	5.67
			(3,0)	5.63
			(4,0)	5.84
			(6,1)	5.80
	4		(3,3)	5.52
			(0,0)	5.23
			(0,3)	5.35
			(2,2)	28.70
	5		(1,0)	15.27
			(1,1)	25.14
			(0,0)	15.22
			(0,1)	15.27
3	3		(0,3)	5.32
			(0,4)	5.49
			(3,0)	5.63
			(4,0)	5.80
			(6,1)	5.78
	4		(3,3)	5.53
			(0,0)	5.23
			(0,3)	5.35
	5		(1,0)	15.18
			(1,1)	26.39
			(0,0)	15.17
			(0,1)	15.17
4	3	A	(0,3)	5.31
		B	(0,4)	5.48
		C	(3,0)	5.63
		D	(4,0)	5.79
		E	(6,1)	5.78
	4	A	(3,3)	5.53
		B	(0,0)	5.23
		C	(0,3)	5.36
	5	A	(1,0)	15.17
		B	(1,1)	26.78
		C	(0,0)	15.16
		D	(0,1)	15.16

(b) Tank image; no changes after the second iteration. Note that one of the roots is a single pixel.

Iteration	Level	Root Label	Coordinates	Value
1	4	A	(1,2)	33.59
		B	(2,1)	34.40
		C	(2,2)	34.59
	5	A	(1,0)	23.49
		B	(1,1)	23.40
		C	(0,0)	20.92
		D	(0,1)	20.96
2	0		(48,63)	43.00
	4		(1,2)	33.45
			(2,1)	33.81
			(2,2)	33.64
	5		(1,0)	23.18
			(1,1)	22.93
			(0,0)	20.37
			(0,1)	20.41
3	0	A	(48,63)	43.00
	4	A	(1,2)	32.96
		B	(2,1)	33.34
		C	(2,2)	33.03
	5	A	(1,0)	22.90
		B	(1,1)	22.33
		C	(0,0)	20.25
		D	(0,1)	20.29

(c) Chromosome image; no changes after the eighth iteration.

Iteration	Level	Root Label	Coordinates	Value
1	2	A	(15,14)	32.82
		B	(2,7)	37.80
		C	(4,10)	37.51
		D	(5,2)	48.73
		E	(6,2)	49.51
		F	(13,0)	53.54
	3	A	(0,1)	35.84
		B	(0,2)	17.16
		C	(0,4)	14.93
		D	(2,0)	23.94
		E	(2,5)	55.38
		F	(2,6)	47.02
		G	(5,3)	53.99
		H	(6,5)	51.50
	4	A	(0,3)	14.03
		B	(1,2)	36.74
	5	A	(1,0)	28.92
		B	(1,1)	19.19
		C	(0,0)	53.83
		D	(0,1)	19.68

(Cont'd. on next page)

2	2			(15,14)	31.31
				(2,7)	37.86
				(4,10)	36.87
				(5,2)	48.48
				(6,2)	48.82
				(6,6)	32.01
				(13,0)	52.93
		3		(0,1)	35.28
				(0,2)	17.32
				(2,5)	55.32
				(2,6)	47.45
				(5,3)	53.77
				(6,5)	52.25
				(7,1)	21.71
		4		(0,3)	14.05
				(1,0)	37.11
		5		(1,0)	28.29
				(1,1)	19.17
				(0,0)	54.19
				(0,1)	19.38
	3	2		(15,14)	29.23
				(2,7)	38.03
				(4,10)	36.65
				(5,2)	48.34
				(6,2)	48.53
				(6,6)	30.35
				(13,0)	52.67
		3		(0,1)	34.21
				(0,2)	17.46
				(2,5)	55.23
				(2,6)	48.17
				(5,3)	53.62
				(6,5)	52.21
				(7,1)	21.39
		4		(0,3)	14.06
		5		(1,0)	27.65
				(1,1)	19.26
				(0,0)	54.15
				(0,1)	19.23
	4	2		(15,14)	26.71
				(2,7)	38.13
				(4,10)	36.50
				(13,0)	52.57
		3		(0,1)	32.97
				(0,2)	17.59
				(2,5)	55.16
				(2,6)	49.19
				(5,3)	53.51
				(5,4)	33.21
				(6,5)	52.12
				(7,1)	21.27
		4		(0,3)	14.07
		5		(1,0)	27.28
				(1,1)	19.36
				(0,0)	54.09
				(0,1)	19.12
	5	2		(2,7)	38.12
				(4,10)	36.22
				(13,0)	52.52
		3		(0,2)	17.70
				(2,5)	55.10
				(2,6)	50.48
				(5,3)	53.44
				(5,4)	32.17
				(6,5)	52.03
		4		(0,3)	14.08
		5		(1,0)	27.04
				(1,1)	19.45
				(0,0)	54.04
				(0,1)	19.06
	6	2		(2,7)	38.03
				(4,10)	35.88
				(13,0)	52.50
		3		(2,1)	28.54
				(2,5)	55.04
				(2,6)	52.32
				(5,3)	53.38
				(6,5)	51.97
		4		(0,3)	14.08
				(1,0)	36.96
		5		(1,0)	26.81
				(1,1)	19.49
				(0,0)	54.01
				(0,1)	19.04
	7	2		(2,7)	37.87
				(4,10)	35.53
				(13,0)	52.49
		3		(2,5)	54.97
				(2,6)	54.00
				(5,3)	53.34
				(6,5)	51.92
		4		(0,3)	14.09
				(1,0)	36.95
		5		(1,0)	26.92
				(1,1)	19.51
				(0,0)	53.99
				(0,1)	19.05
	8	2		(2,7)	37.65
				(4,10)	35.16
				(13,0)	52.49
		3		(2,5)	54.93
				(2,6)	54.69
				(3,1)	48.04
				(5,3)	53.31
				(6,5)	51.89
		4		(0,3)	14.10
				(1,0)	36.96
		5		(1,0)	26.95
				(1,1)	19.51
				(0,0)	53.99
				(0,1)	19.05
	9	2	A	(2,7)	37.36
			B	(4,10)	34.75
			C	(13,0)	52.50
		3	A	(2,5)	54.90
			B	(2,6)	54.94
			C	(3,1)	48.07
			D	(5,3)	53.29
			E	(6,5)	51.86
		4	A	(0,3)	14.10
			B	(1,0)	36.96
		5	A	(1,0)	26.84
			B	(1,1)	19.49
			C	(0,0)	53.99
			D	(0,1)	19.05